

8N.09/652365

FIG. 1

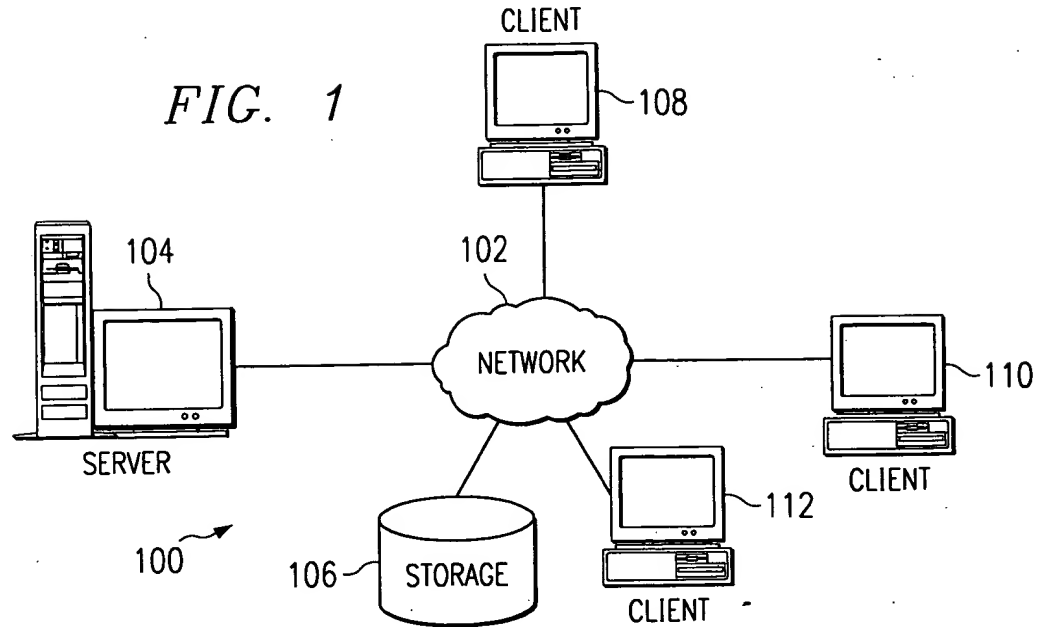
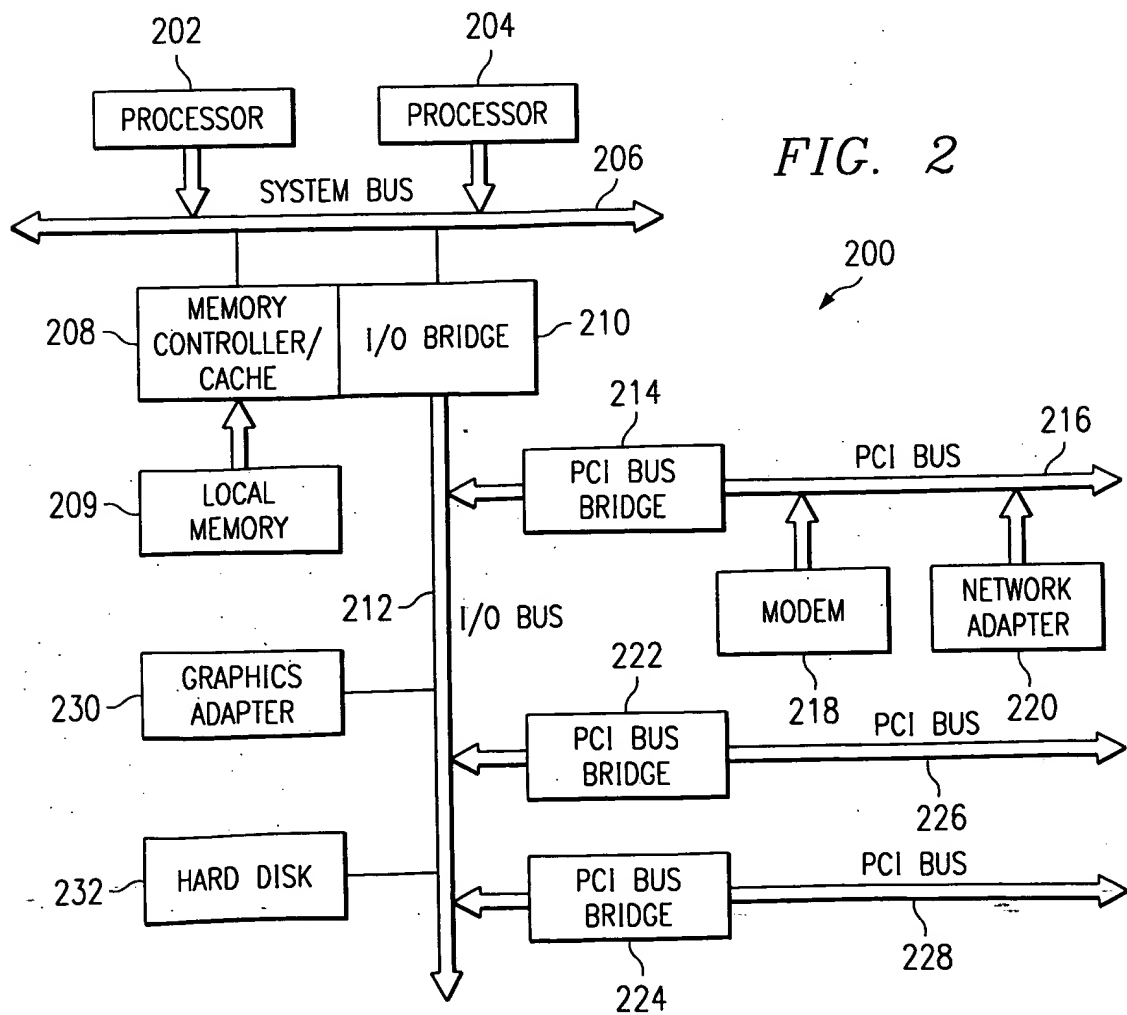


FIG. 2



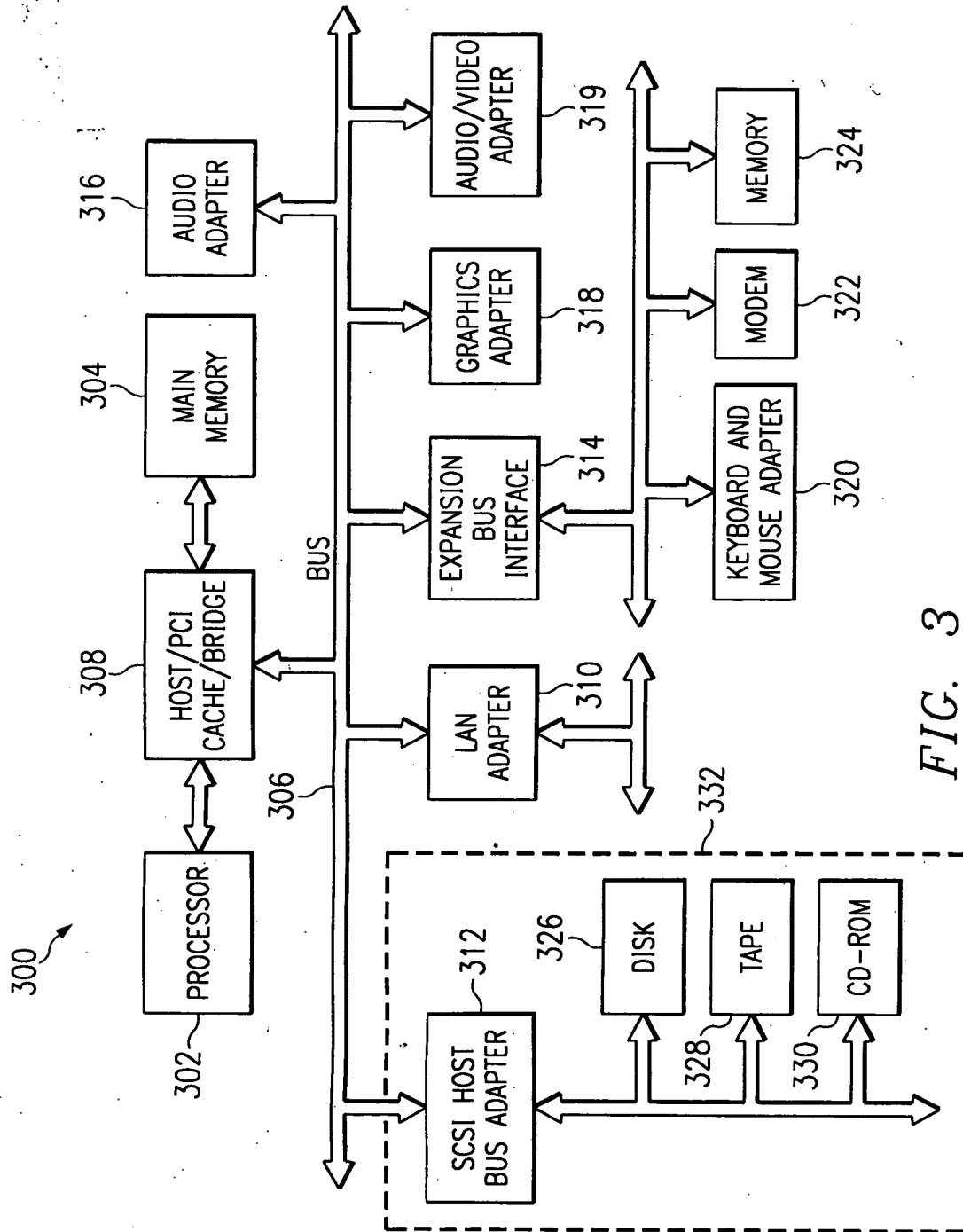


FIG. 3

FIG. 4

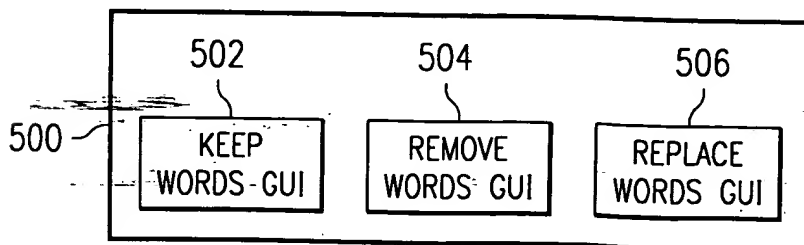
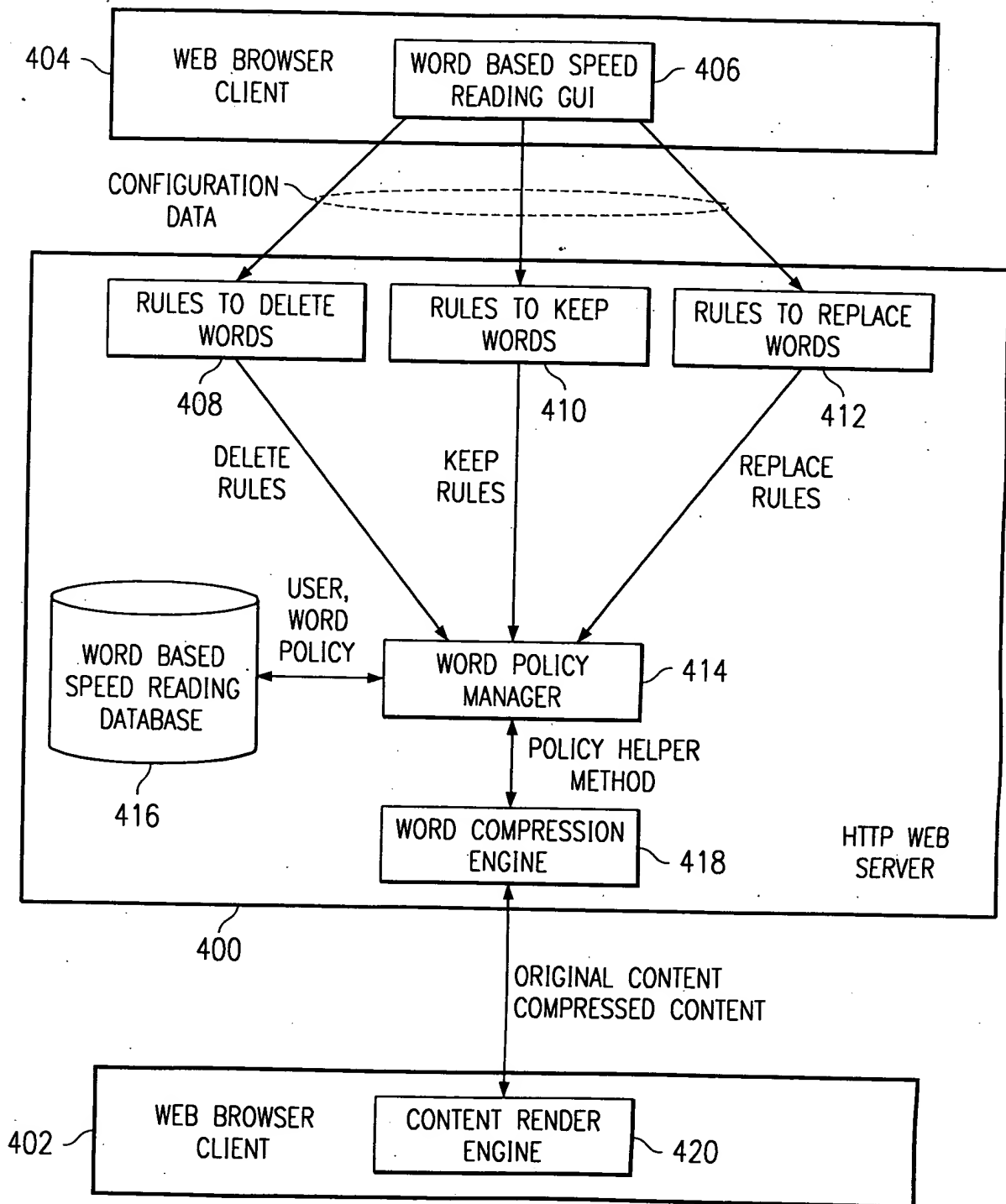
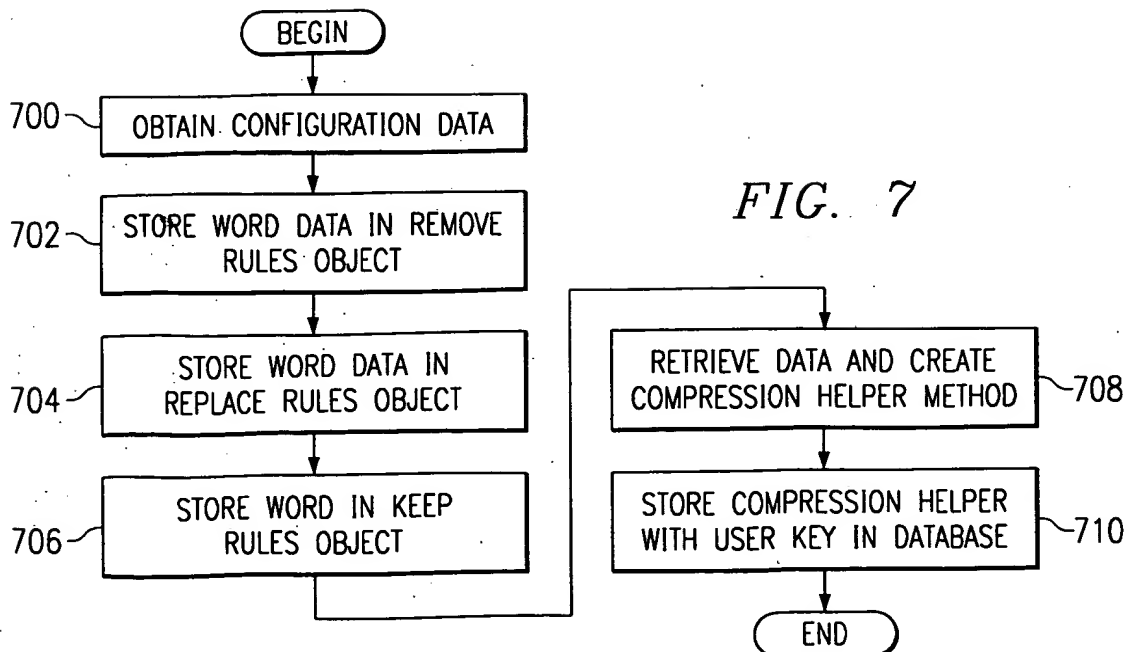
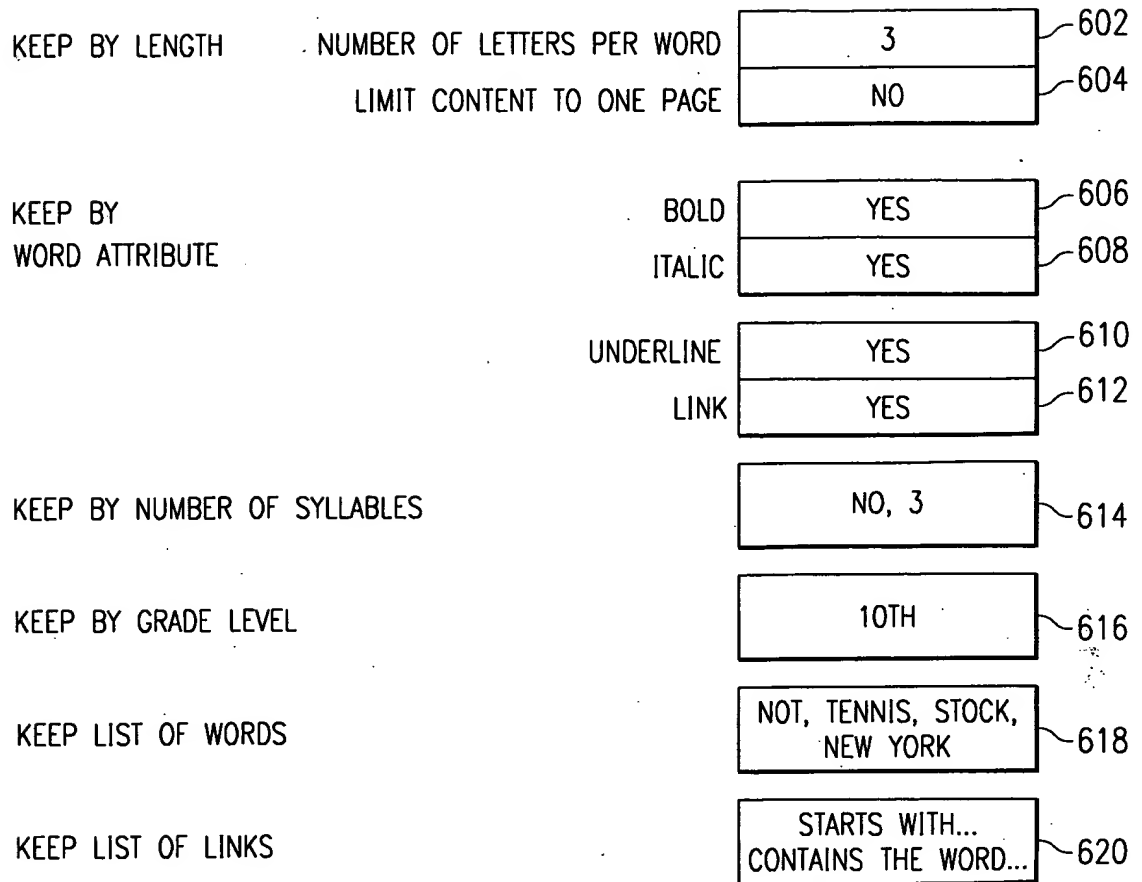


FIG. 5

FIG. 6



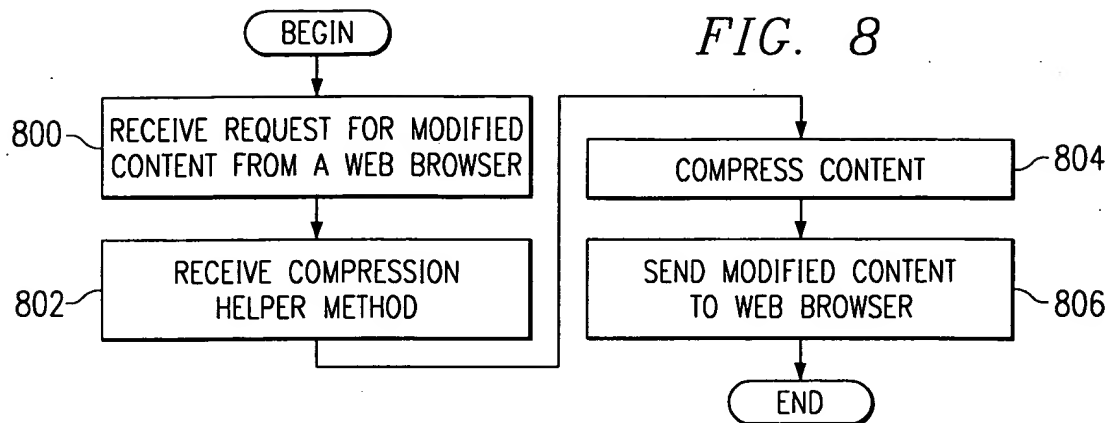


FIG. 9A

DeleteWord { // used if delete words defined in GUI

900

//init constructor

DeleteWord (int length, int syllable, int Difficulty, int attribute)

//data

int byLength;

int bySyllable;

int byDifficulty;

int byAttribute;

Vector removeFullWords;

Vector removeStartsWithWords;

//methods

void setFullWord (String word) { }

void setStartsWith (String word) { }

boolean delete Word (String word) {

 //compare length, difficulty, attributes

 //compare with removeFullWords list

 //compare with removeStartsWithWords list

 //return true or false

void setWord (String word) {

 //used by the GUI to add words to delete (or extended by Keep Class below)

 //add to Vector

}

void setStartsWithWord (String word) {

 //used by the GUI to add words

 //add to Vector

}

FIG. 9B

```
ReplaceWords { // used if replace words defined in GUI
    //data
    Hashtable wordsToBeReplaced
        //key=word to be replaced
        //value=replacement word
    //methods
    boolean replace (String word) {
        //check hash to decide return true
    }
    void setReplacement (String wordToBeReplace, replacement) {
//used by the GUI to add words
    }
    String getReplacement (String word) { }
}
KeepWords { // used if keep words defined in GUI
    extends DeleteWords

    //init constructor
    KeepWords (int length, int syllable, int Difficulty, int attribute)

    //methods

    boolean keepWord (String word) {
        //compare length, difficulty, attributes
        //compare with removeFullWords list
        //compare with removeStartsWithWords list
        //return true or false.
    }
}
```

FIG. 10

1000

```

CompressionHelper {

//methods
String getReplacement (word){ };

    int getNumSyllables (String word) { return numberOfSyllables }
    int getWordLength (String word) { return wordLength }
    int getDifficulty (String word) { return GradeLevelDifficulty }
    int getAttributes (String word) //bold=1, underline=2, italic=3, etc.
    { return VectorOfAttributes

//data
boolean isPartOfWordRemoveList = DeleteWords.deleteWord (word);
boolean isPartOfWordKeepList = KeepWords.keepWord (word);
boolean isPartOfWordReplaceList = ReplaceWords.deleteWord (word);

//init Constructors
CompressionHelper(String word) { };
CompressionHelper(String [ ] words) { };

}
    
```

FIG. 11

1100

```

//String getModifiedContent (String OriginalContentInFileFormat) {
    //create InputStream from OrginalContentInFile
    //create InputStream for ModifiedContentOutputFile
    //loop through all words
        ifWordOnDeleteList //create CompressionHelper classes with each word to be
analyzed in parallel with reading the unmodified file content. After caching the
compression helpers away, the boolean flags can be used to determine how the
modified content is rendered (word removed, word replaced, word remains intact).
and NOT on isWordOnKeepList OR isWordOnReplaceList
            //delete word
            //break next word
        else IfOnKeepList AND NOT on WordReplaceList
            //break next word
        else //OnWordReplacelist
            //replace word

    -Write result to ModifiedContentOutoutFile
}
    
```